

~~EPICS Base development at NSLS2~~
Unit-testing Records

Michael Davidsaver
BNL NSLS2

What is a unit-test

- ... according to Michael
- Isolated
 - No interactions with the outside world
 - Repeatable (by resetting state)
- In process
 - Inspection of memory
- Fast
 - Run lots of test cases quickly

What to unit-test?

- When does unit-testing made sense?
 - When all state in software
 - subRecord and asubRecord
 - Complex DB logic
- Hardware?
 - HW is stateful
 - Isolation often not possible
 - Other forms of testing more appropriate

Unit-testing in 3.15

- dbUnitTest.h
 - Added in 3.15.1
- Allows unit-testing of Database records
 - Records and support code initialized
 - Scanning threads started
 - initHooks run
 - ...
- Isolated
 - CA_LINKs disabled
 - RSRV not started

Unit-testing in 3.15 (2)

- Shutdown of core IOC
 - Scan/callbacks threads stopped
 - Record structs free'd
- Not complete shutdown
 - No hooks for record/device support cleanup
 - eg. Array fields of aSubRecord are leaked
- Local CA_LINKs
 - Never connect (planned feature for 3.16)

New testing API example

```
static void doTest(void) {
    testdbPrepare();

    testdbReadDatabase("utest.dbd", NULL, NULL);
    utest_registerRecordDeviceDriver(pdbbase);

    testdbReadDatabase("dut.db", NULL, "N=test:,A=42");

    testlocInitOk();

    /* Test stuff here */
    testOk1(...);
    testdbPutFieldOk("rec:name", DBR_LONG, 1);
    testdbGetFieldEqual("rec:name", DBR_LONG, 1);

    testlocShutdownOk();
    testdbCleanup();
}
```

Full example at <https://github.com/mdavidsaver/utestdemo>

Future additions

- Extend testing API
 - Convenience wrappers for common operations
- Expand testable parts of Base (and beyond)
 - Local CA and CA_LINKs
 - SNL program + Records
- Cleanup hooks for record/device support

Full example at <https://github.com/mdavidsaver/utestdemo>